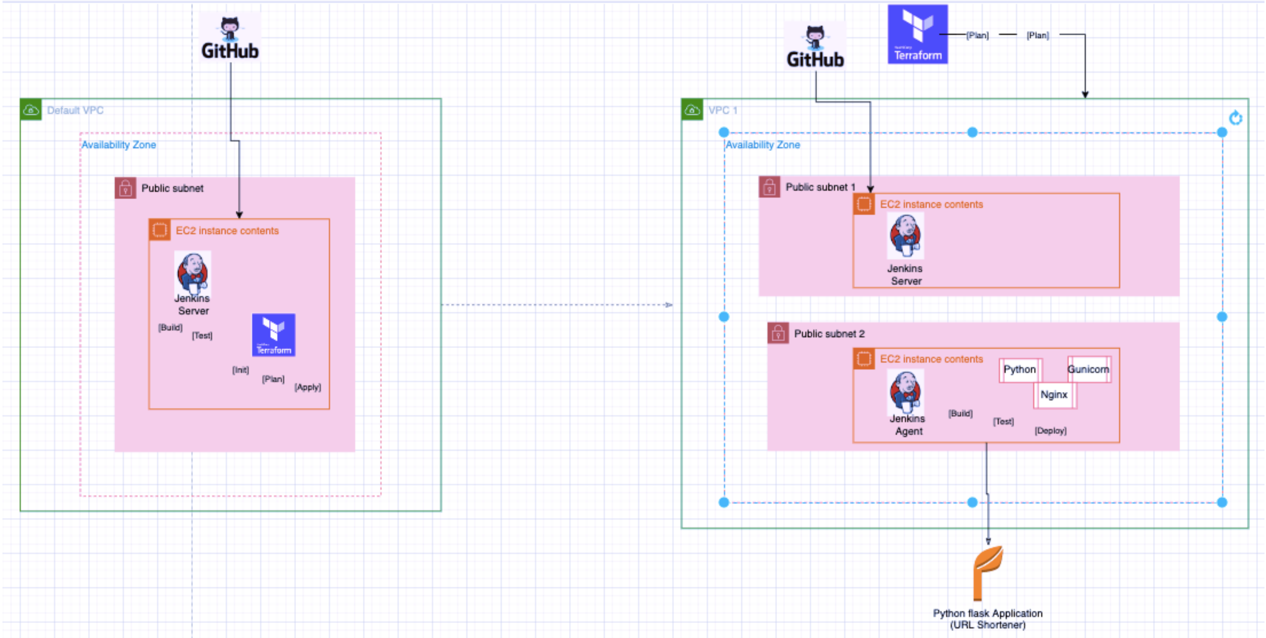
DEPLOYMENT #4



Purpose

This deployment first used a Jenkins pipeline on a default VPC to run terraform then terraform was used to build and configure the infrastructure on AWS for a Jenkins pipeline to deploy a URL shortener web application.

Tools/services and software stacks used:

- GitHub repository
- AWS VPCs
- AWS EC2s
- Terraform
- Jenkins (server and agent)
- Web server (nginx)
- Web server (gunicorn)

GitHub

- The code to build the url-shortner web application was stored in a GitHub repository.
- The source repository was first forked to enable a copy of the original repository to reside in my GitHub repository.
- Jenkins connected to my GitHub repository to initiate the build process in both parts of the deployment

AWS VPCs

- Two VPCs were created .
- The first VPC was created manually and was used to house the Jenkins server and terraform.

- The second VPC was created by the Infrastructure as Code software, terraform.

AWS EC2s

- One EC2 instance was launched in a public subnet in the manually created VPC.
    - It ran the Jenkins server with port 80 (HTTP), port 22 (SSH) and port 8080 (Jenkins) open for inbound traffic.
- Two EC2 instances were launched in two separate public subnets in the VPC built by terraform
    - One ran the Jenkins agent with port 80 (HTTP), port 22 (SSH) and port 5000 (TCP/IP) open for inbound traffic.
    - The other ran the Jenkins server with port 80 (HTTP), port 22 (SSH) and port 8080 (Jenkins) open for inbound traffic.
    -
- Python3-pip, python3-10-venv, default-jre and nginx packages were installed in the virtual environments to update them with the necessary libraries and dependencies.

Terraform

The Infrastructure as Code (IaC) tools, terraform was first tested and then used to build, change and manage the infrastructure set up and configured to deploy a web application

Jenkins

- The Jenkins server was first set up on the default VPC.
    - A multibranch pipeline build was selected with GitHub being the designated branch source.
    - The server would automatically discover, manage, and execute pipelines for branches which contain a Jenkinsfile in source control.
    - The server ran the build and test for the testing of the terraform application.
- Another Jenkins server was set up by terraform in VPC1 and also configured for a multibranch pipeline build.
- A Jenkins agent was also set up by terraform in VPC1. It was then configured and connected to the Jenkins server to only build jobs with the label/name **awsDeploy**
- Jenkins agent ran the build, test, and deploy stages of the url-shortner app successfully.

Web server

- Webserver nginx, accepts request, takes care of domain logic, and handles http connections requests rerouted to listen on port 5000 instead of 80.

Web server

- Gunicorn, a web server gateway interface (WSGI) ensures that the web server and the python flask application **URL shortener** can talk to each other on port 8000.